SC25 BoF
Tuesday - Nov 18, 2025
12:15pm - 1:15pm CT
Room 130

IRI Interfaces at Work: Prototypes, Progress, and Community Feedback

Bjoern Enders
NERSC

John MacAuley
ESnet

Paul Rich
ANL

Juan Pablo Dorsch
CSCS

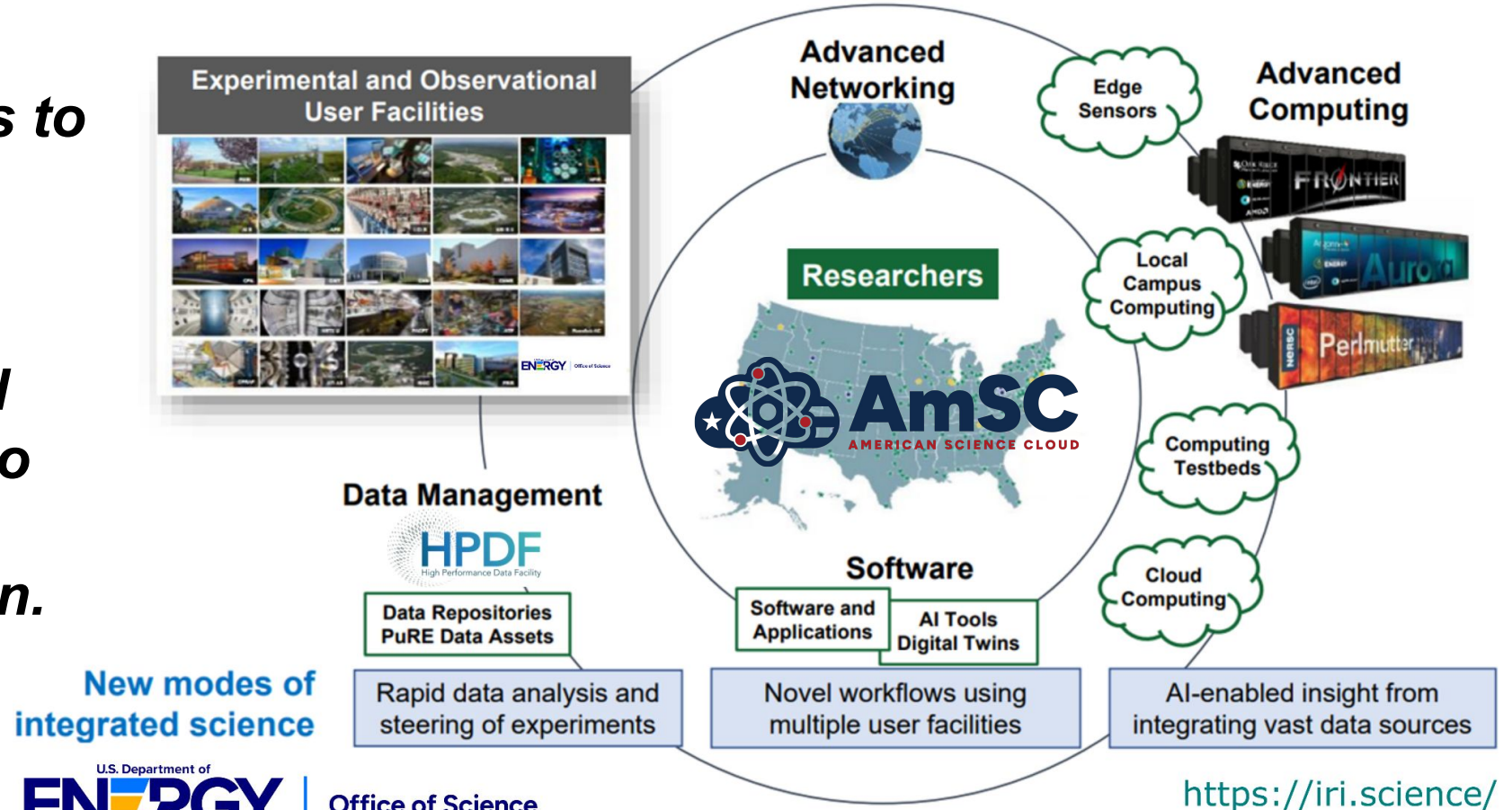
Tiffany Connors
NERSC

Eli Dart
ESnet

SC25 ST. LOUIS hpc ignites.

DOE's Integrated Research Infrastructure (IRI) Vision

To empower researchers to meld DOE's world-class research tools, infrastructure, and user facilities seamlessly and securely in novel ways to radically accelerate discovery and innovation.



IRI technical work is coordinated via Technical Subcommittees, with membership across ASCR facilities

IRI-ABA Practice Area	IRI Program Area	Status
Resource Co-Operations	IRI Allocations Program/ Scheduling/Preemption Technical Subcommittee	Under development; Under development
Cybersecurity and Federated Access	TRUSTID Technical Subcommittee	Launched!
User Experience	Outreach/Engagement Technical Subcommittee	Launched!
Workflows, Interfaces & Automation	Interfaces Technical Subcommittee	Launched!
Scientific Data Lifecycle	Data Movement Technical Subcommittee	Under development
Portable / Scalable Solutions	Software Deployment and Portability	Under development

Slide adapted from Debbie Bard

Agenda

- 12:15 Welcome
  - NERSC - Bjoern Enders
- 12:20 A one-year review of the IRI interfaces subcommittee: Current status and achievements
  - ESnet - John McAuley
- 12:30 Panel discussions and Open Q&A with IRI guest stakeholders:
  - TrustID - Tiffany Connors
  - User Engagement - Eli Dart
- 12:40 Guest presentation
  - CSCS - Juan Pablo Dorsch
- 12:50 Future roadmap and design plans for IRI APIs
  - ANL - Paul Rich
- 1:00 Closing panel discussion

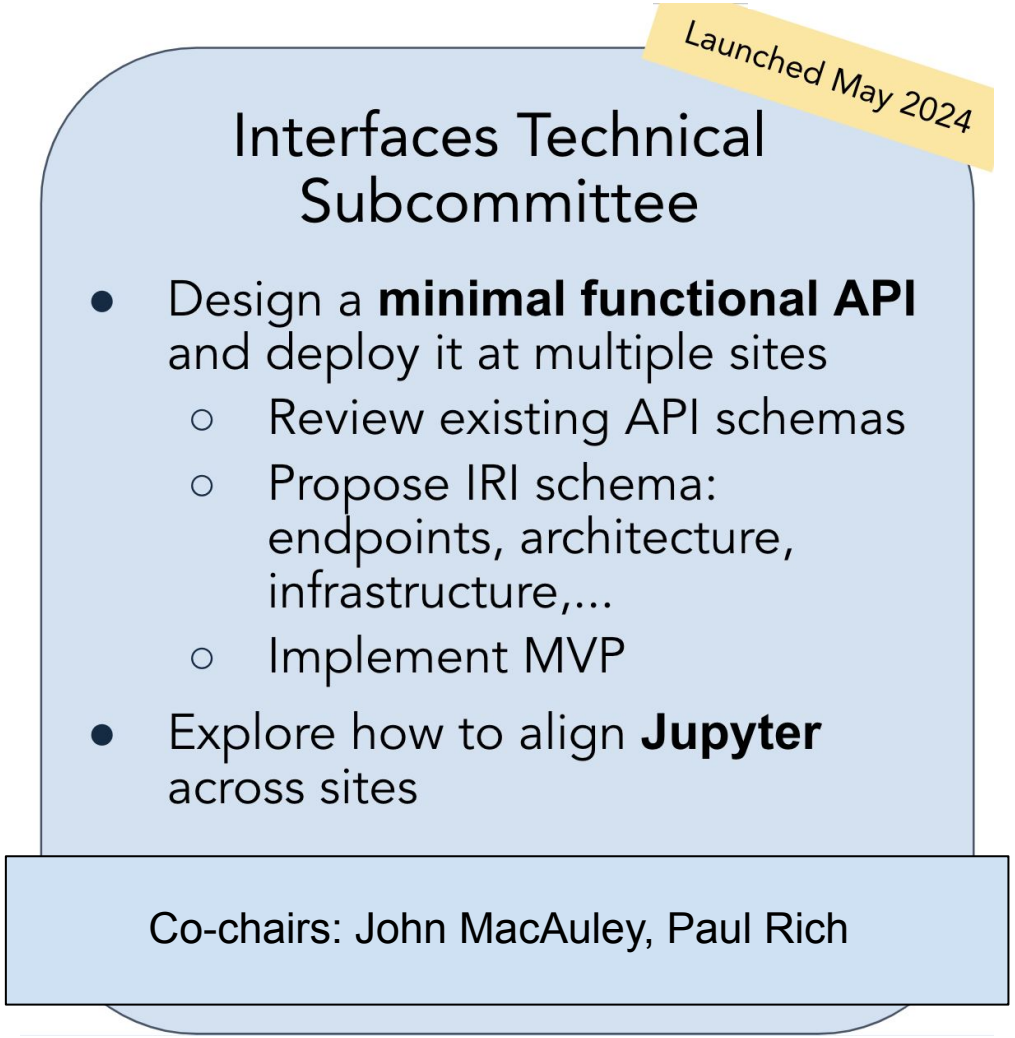
IRI Presentations

A One-Year Review of the IRI Interfaces Subcommittee: Current Status and Achievements

John MacAuley
Energy Sciences Network

IRI Interfaces Charter

- Gather requirements, define use cases, and specify unified interfaces for the DOE ASCR Facilities.
- Tasked with enhancing the "user experience" for interaction and management of complex workflows across ASCR facilities
  - Goal is to facilitate a more cohesive and efficient research infrastructure.
- Consider applicable standards and build upon the work of other organizations where available to deliver a reference IRI interface implementation.



IRI Development – How & What

- The How
  - The Interfaces Subcommittee co-designs the IRI specification
  - Ideas are discussed, prototyped, and battle-tested before they land in the spec
  - Decisions are not sacred - we revisit and refine them as we learn
- The What
  - HTTP/JSON APIs shaped by pragmatic REST principles
  - An OpenAPI specification serves as the formal contract and documentation
  - Reference implementations jump-start adopters and boost development velocity
  - Multiple independent implementations validate interoperability and the spec itself

IRI Facility API Specification*

ExaWorks PSI/J

Facility

- Facility
- Site
- Location
- Resource

Status

- Incident
- Event
- Resource

Allocation (Draft)

- Project
- UserAllocaton
- ProjectAllocation
- AllocationEntry
- Capability
- Resource

Compute (Draft)

- Job
- Resource

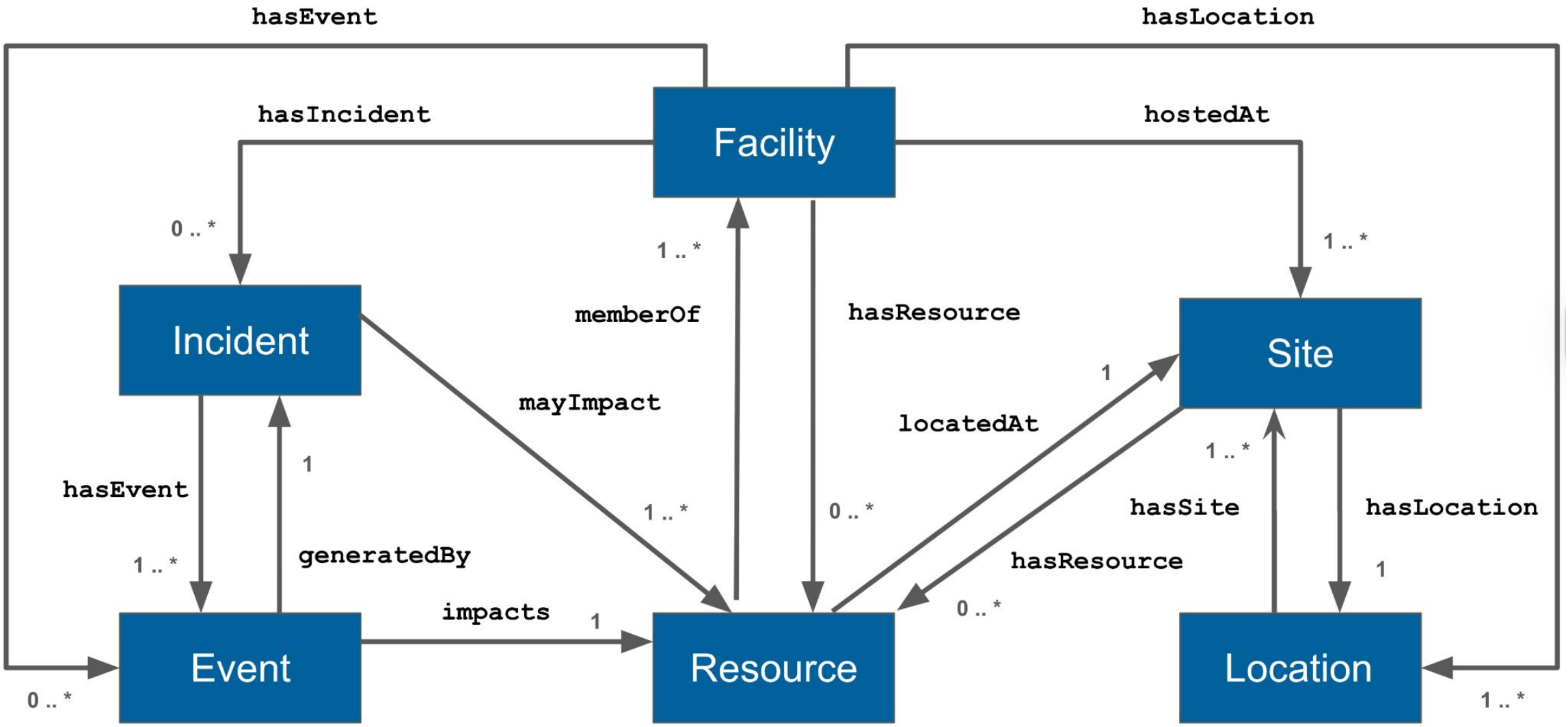
FIRECREST

Filesystem (Draft)

- File
- Path
- Resource

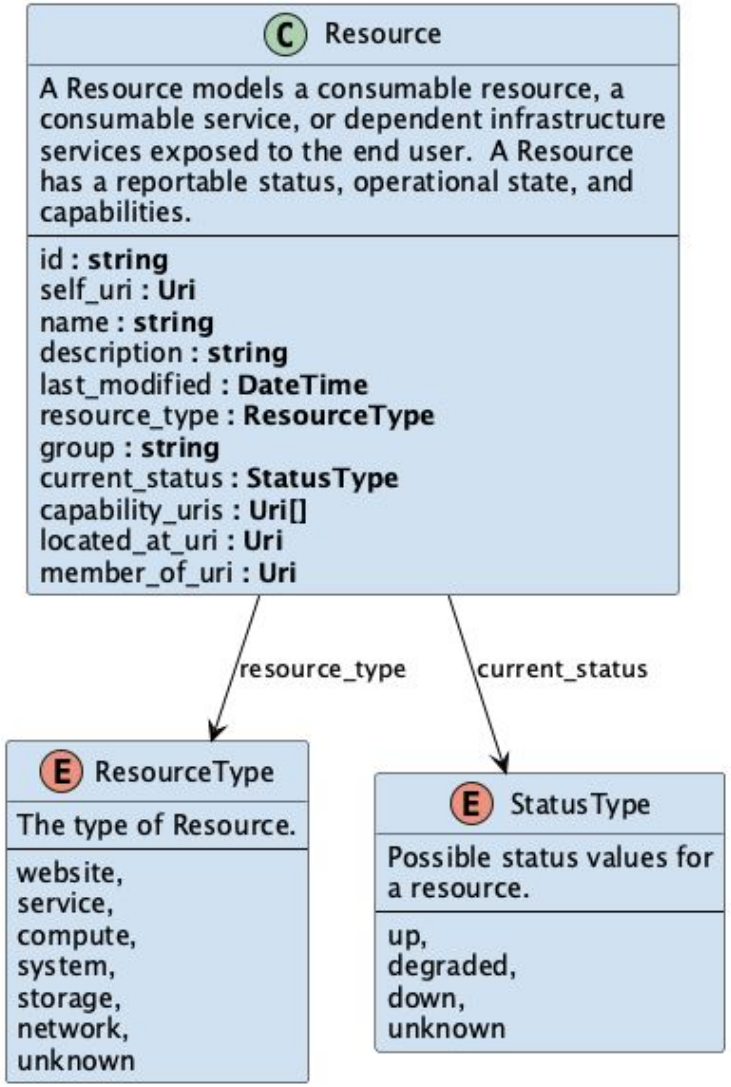
* https://github.com/doe-iri/iri-facility-api-docs

Facility Model



SC25 hpc ignites.

Resource Example




```
{
  "id": "fa7a5eba-d498-4dad-8ebb-a9f67110ab62",
  "self_uri": "http://localhost:8081/api/v1/status/resources/fa7a5eba-d498-4dad-8ebb-a9f67110ab62",
  "name": "Compute Nodes",
  "description": "The Perlmutter computer compute nodes",
  "last_modified": "2025-10-20T02:02:31.000Z",
  "group": "Perlmutter",
  "resource_type": "compute",
  "current_status": "degraded",
  "capability_uris": [
    "http://localhost:8081/api/v1/account/capabilities/bfa7bb6e-8baf-468a-a92e-2679c04976b8",
    "http://localhost:8081/api/v1/account/capabilities/c6c1c91d-4821-453a-b3a7-ebecc9581351"
  ],
  "located_at_uri": "http://localhost:8081/api/v1/facility/sites/ce2bbc49-ba63-4711-8f36-43b74ec2fe45",
  "member_of_uri": "http://localhost:8081/api/v1/facility"
}
```

Status Model

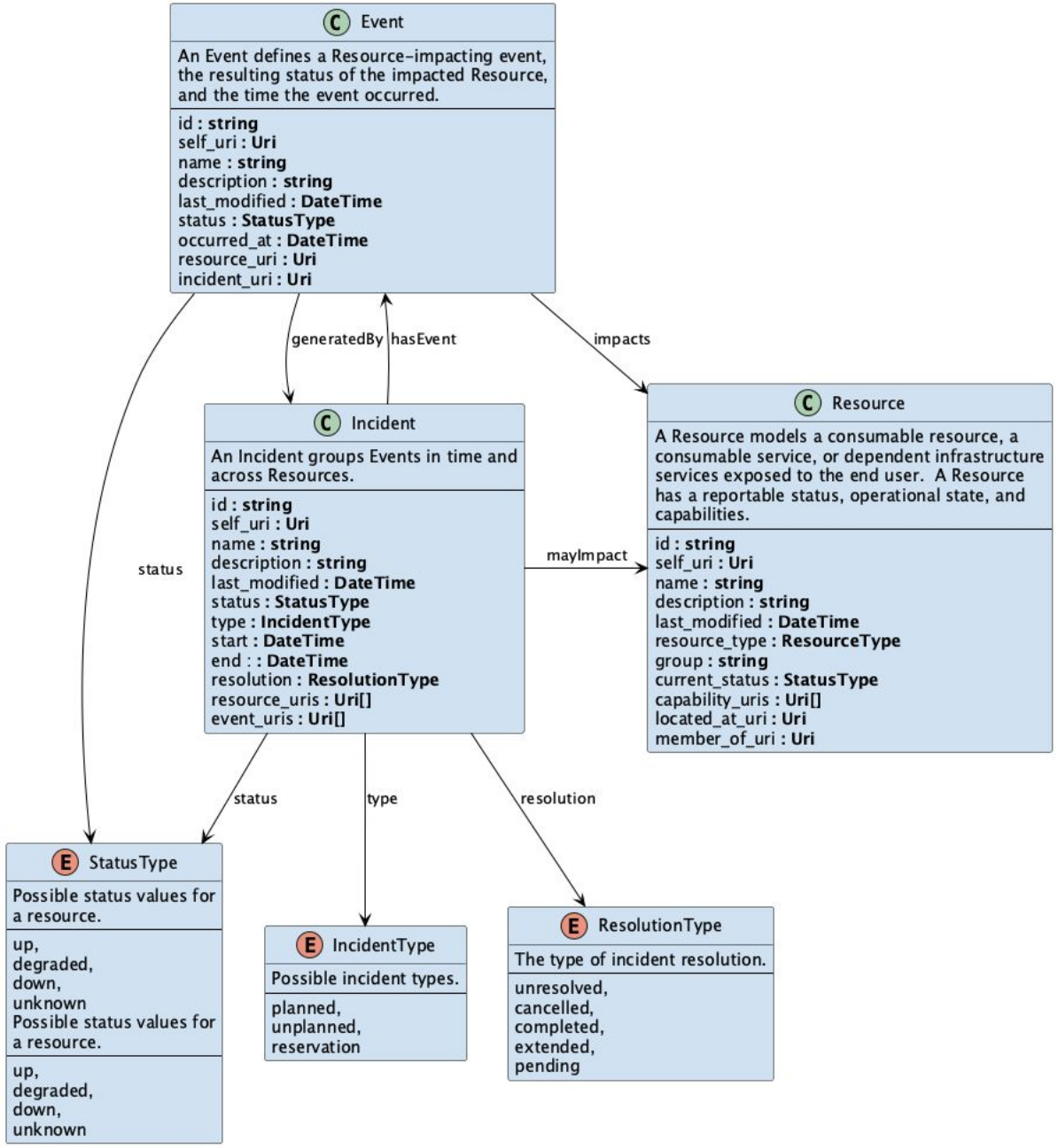
An Incident represents a discrete occurrence, planned or unplanned, that actually or potentially affects the availability, performance, integrity, or security of one or more Resources at a given Facility.

Incidents serves as a high-level grouping construct for aggregating and tracking related Events over time and across multiple system components acting as higher-level status records for users and operators.

An Event is a discrete, timestamped occurrence that reflects a change in the state, condition, or behavior of a Resource, typically within the context of an ongoing Incident.

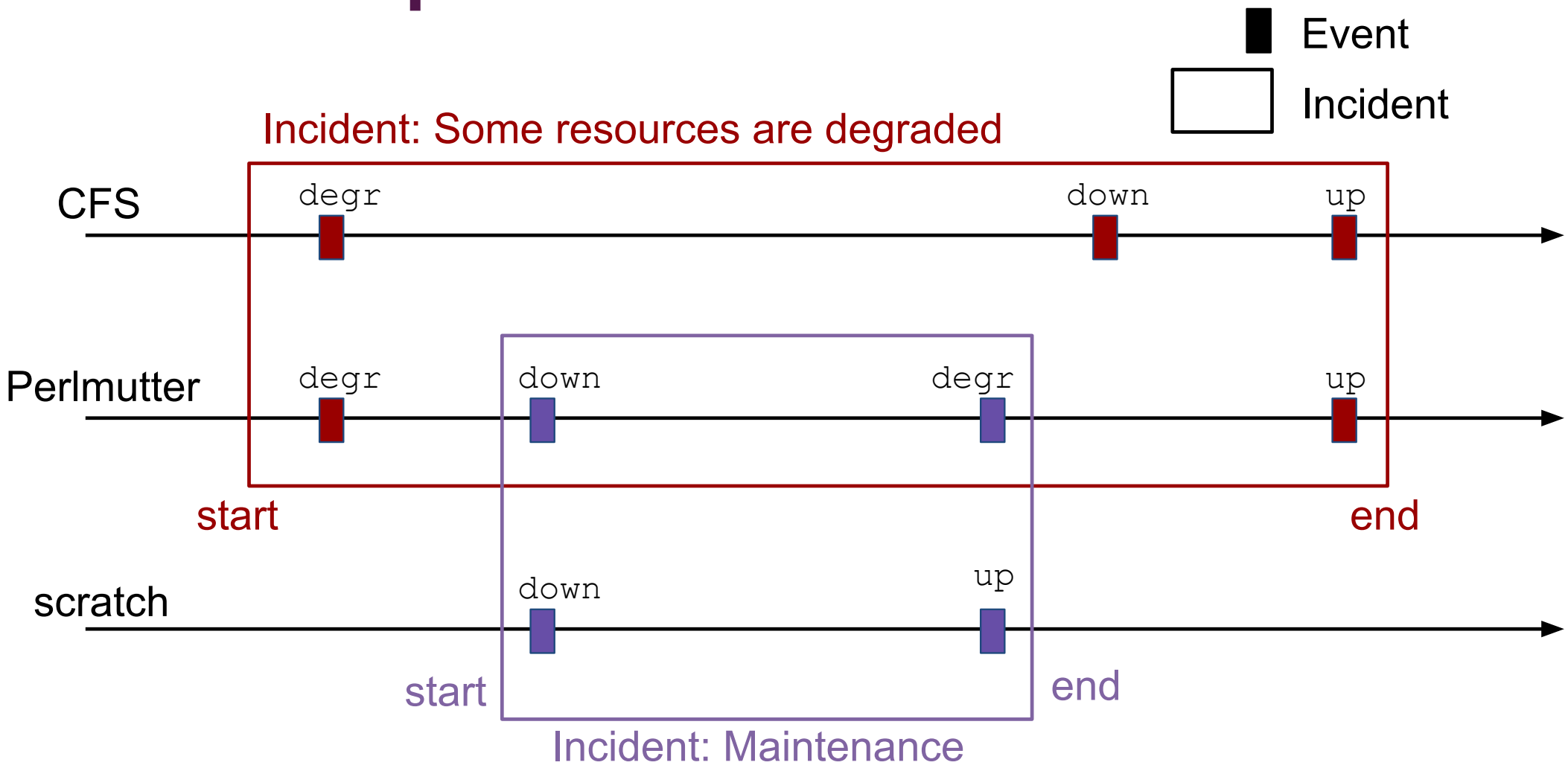
Events provide the fine-grained details necessary to understand the progression and impact of an Incident (e.g., transitions between up, degraded, and down).

Events are immutable records useful for audits and timelines.



SC25 hpc ignites.

Status Example



e.g. https://api.iri.nersc.gov/nersc/api/current/status/incidents/65e8ea08-a937-410a-983f-19aabdaa1837

Year Two: Status Update

Delivered

- Facility and Status API (unauthenticated).
- Draft of Allocation, Job, and Filesystem API (authenticated).
- IRI Tokens: Format Proposal.

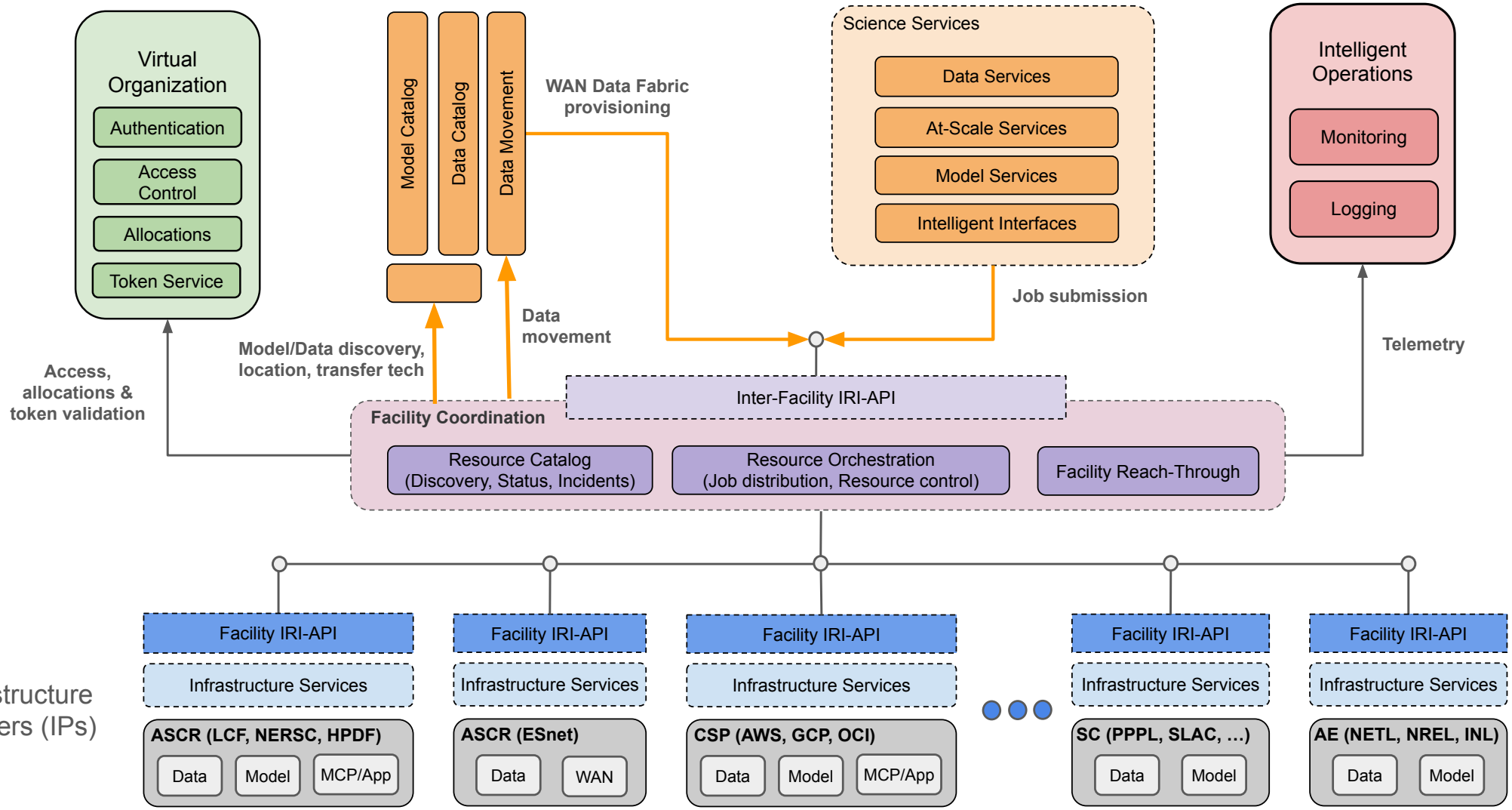
Facility implementations

- NERSC, OLCF, ALCF, and ESnet developing unauthenticated solutions for Facility and Status API.

Impacts of American Science Cloud (AmSC)

- Redefined one-year roadmap to fit needs of AmSC deliverables.
- Working closely with AmSC peers to understand changing / re-prioritization of requirements.
- Monitoring of defined deliverables by AmSC project managers.

Expanding IRI Ecosystem



Panel Q&A

<——————— IRI interfaces ———————>

Bjoern Enders
NERSC

John MacAuley
ESnet

Paul Rich
ANL

Juan Pablo Dorsch
CSCS

IRI TrustID

Tiffany Connors
NERSC

IRI Outreach

Eli Dart
ESnet

SC25 ST. LOUIS hpc ignites.

Invited Presentation

FirecREST v2: a common interface for HPC and AI workflows

Juan Pablo Dorsch
Swiss National Computing Center

The FirecREST journey

- FirecREST is an open-source RESTAPI for advanced automated workflows on HPC
- Based on web standards, modularity, and abstraction; CSCS aims to provide a generic API interface that can be used by any HPC Centre

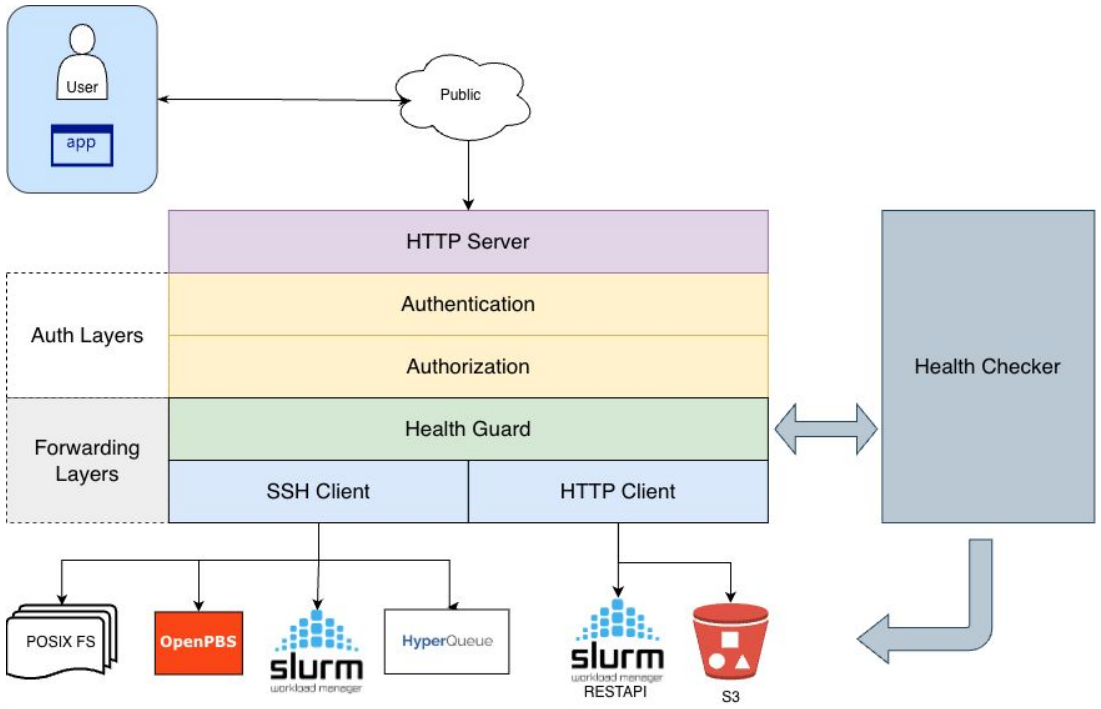
- FirecREST version 1 started its design in 2018
- It was publicly released in late 2019
- Deployed on production in 2021 at CSCS

As announced 1 year ago in SC24, after 5 years of development, FirecREST team decided, based on feedback from users and new use cases, that an evaluation towards an improved interface was required

Improving FirecREST API

- FirecREST version 2 presents a layered architecture that allows to interact with more flexibility with HPC and AI services

- Dedicated clients for HTTP requests and SSH commands execution allows seamless integration with multiple schedulers and storage vendors



SC25 hpc ignites.

Improving FirecREST API

- Web server
  - Framework and libraries selection for full Async performance
    - FastAPI, Uvicorn, AsyncIO, AsyncSSH
- SSH Authorization
  - Introducing the ***SSH Connection Pool***, that outperforms v1 on command execution
  - This new approach re-uses an SSH connection to execute several commands on a row.
- Simplified interface
  - For scheduler abstraction follows the guidelines of SLURM RESTAPI
  - One `filesystem` endpoint, divided into `transfer` and `ops` resources
  - `status` endpoint retrieves status of systems, schedulers, and filesystems

Testing the new version under real use case conditions

AiiDA is a workflow manager that provides high-throughput calculations on HPC

https://github.com/aiidateam/aiida-firecrest

- The use case selected was I/O bound operations to measure latency
  - Up to 1000 small files (1KB) were generated on the server and were retrieved via FirecREST using v1 and v2 endpoints.
  - API requests were made with AsyncClient of the httpx Python library
  - The CSCS' Alps HPE-Cray system Eiger was used
  - Requests were made from a public network

SC25 hpc ignites.

Testing the new version under real use case conditions

Number of files	FirecREST v1	FirecREST v2	FirecREST v2 (w/SSH connection pool)
1	1.5 ± 0.1 [s]	0.8 ± 0.02 [s]	**0.4 ± 0.3 [s]**
10	13.7 ± 1.1 [s]	2.7 ± 0.05 [s]	**0.5 ± 0.2 [s]**
100	129.6 ± 1.5 [s]	19.5 ± 3.4 [s]	1.5 ± 0.7 [s]
1000	*Too long (> 1000 [s])*	176.3 ± 4.5 [s]	**15.5 ± 8.9 [s]**

Testing the new version under real use case conditions

Number of files	FirecREST v1	FirecREST v2	FirecREST v2 (w/SSH connection pool)
1			0.4 ± 0.3 [s]
10	13.7 ± 1.1 [s]	2.7 ± 0.05 [s]	
100	129.6 ± 1.5 [s]	19.5 ± 3.4 [s]	1.5 ± 0.7 [s]
1000	*Too long (> 1000 [s])*	176.3 ± 4.5 [s]	15.5 ± 8.9 [s]

These results permitted AiiDa to provide a **High Throughput I/O workflow**, and, in general, an impressive improvement to FirecREST workflows, overall

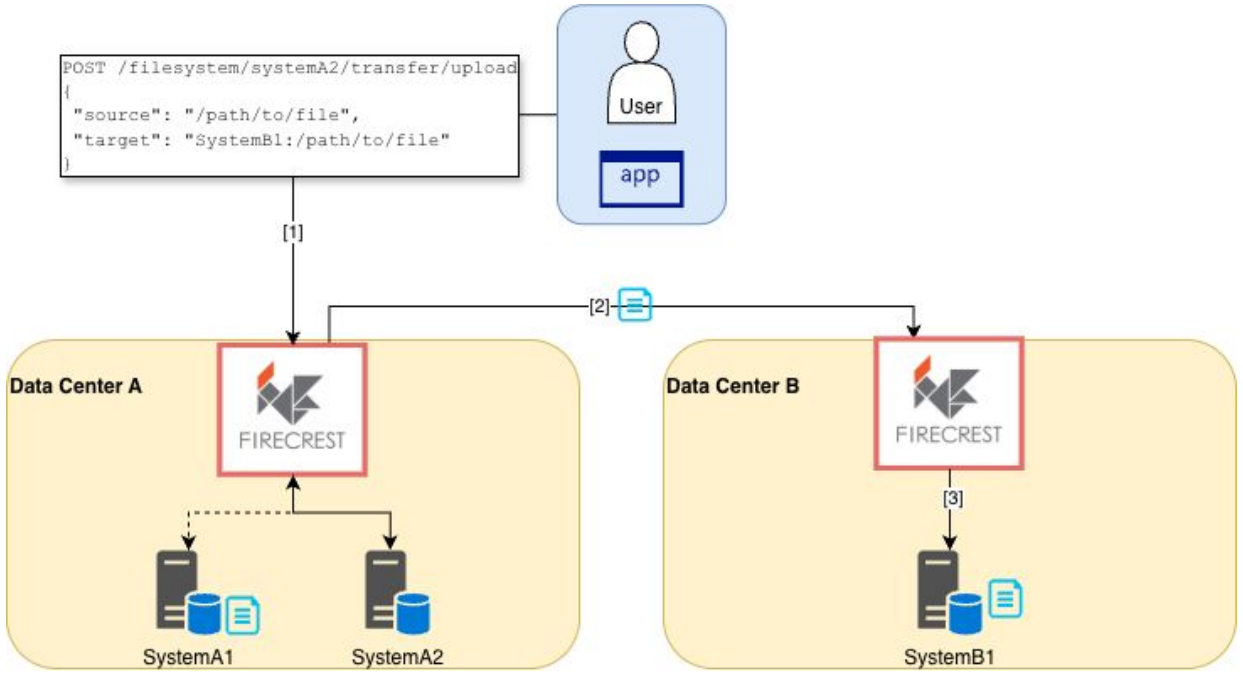
~80x faster than v1

SC25 hpc ignites.

Future Roadmap

- Large Data Transfer abstraction
  - Object Storage staging area DONE
  - Data streaming Testing 2025Q4
  - Globus Connect Backlog 2026Q1

- API to API Data Transfer Protocol



SC25 hpc ignites.

Future Roadmap

Is it possible to get rid of all the SSH connections to the backend infrastructure (filesystems and schedulers)?

At first glance, **yes**:

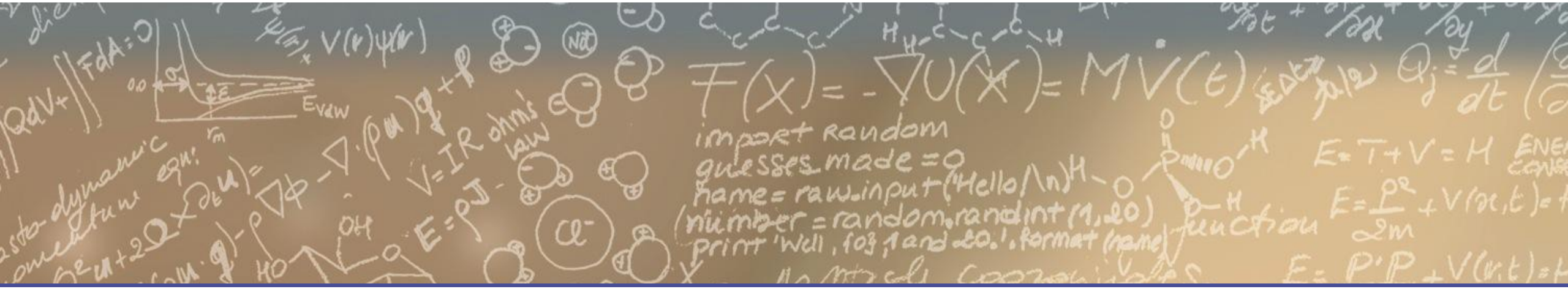
- For schedulers: SLURM API seems stable. Would other vendors provide a similar interface?
- For large data transfers: S3 and Globus. Some filesystem's vendors provide S3 interface or API interface.

However

- There must be APIs for filesystem operations (listing items, creating directories, etc).
- The main challenge remains in orchestrating authentication and authorization across APIs

More on FirecREST

- **FirecREST: github.com/eth-cscs/firecrest-v2**
- API Reference: eth-cscs.github.io/firecrest-v2/openapi
- pyFirecREST: github.com/eth-cscs/pyfirecrest
- FirecREST Web UI: eth-cscs.github.io/firecrest-ui
- Join our community on Slack: firecrest-community.slack.com
- Contact us: firecrest@cscs.ch



Thank you for your attention.

IRI Presentations

IRI Roadmap Update

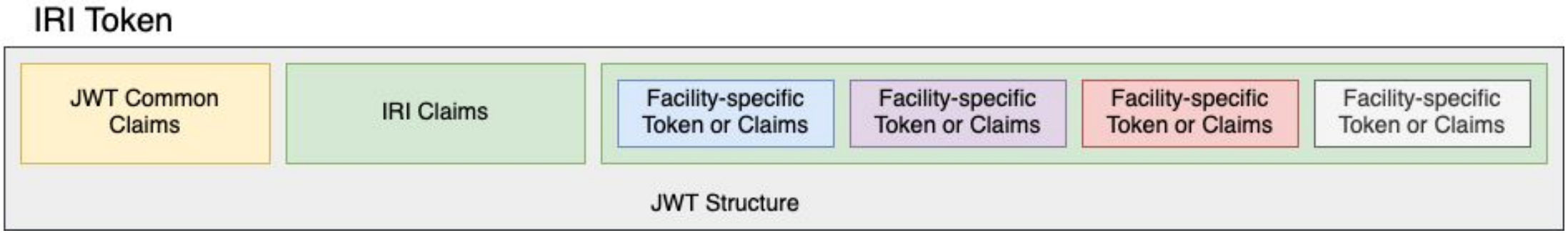
Paul Rich
Argonne National Lab

Roadmap – In-Progress

- IRI Status APIs
  - Official documentation of Facility, Site, and Resource discovery APIs
    - Java and Python Implementations
    - https://github.com/doe-iri
- IRI Authenticated APIs
  - Token-based Authentication
  - Job Script Submission
  - User Accounting Information
  - Local Filesystem Operations
  - Parallel development of reference implementations and documentation
- New Endpoints
  - Use by Commercial Partners via American Science Cloud

Roadmap – Authentication

- IRI JWT
  - Initially IRI JWT is tagged with facility-specific tokens
  - Over time can be replaced by IRI JWT itself
- AmSC efforts expected to accelerate integration and policies
- Why
  - Allows more rapid spin-up of authenticated interface as federation worked out
  - Path for faster onboarding of new facilities as they are introduced to the ecosystem

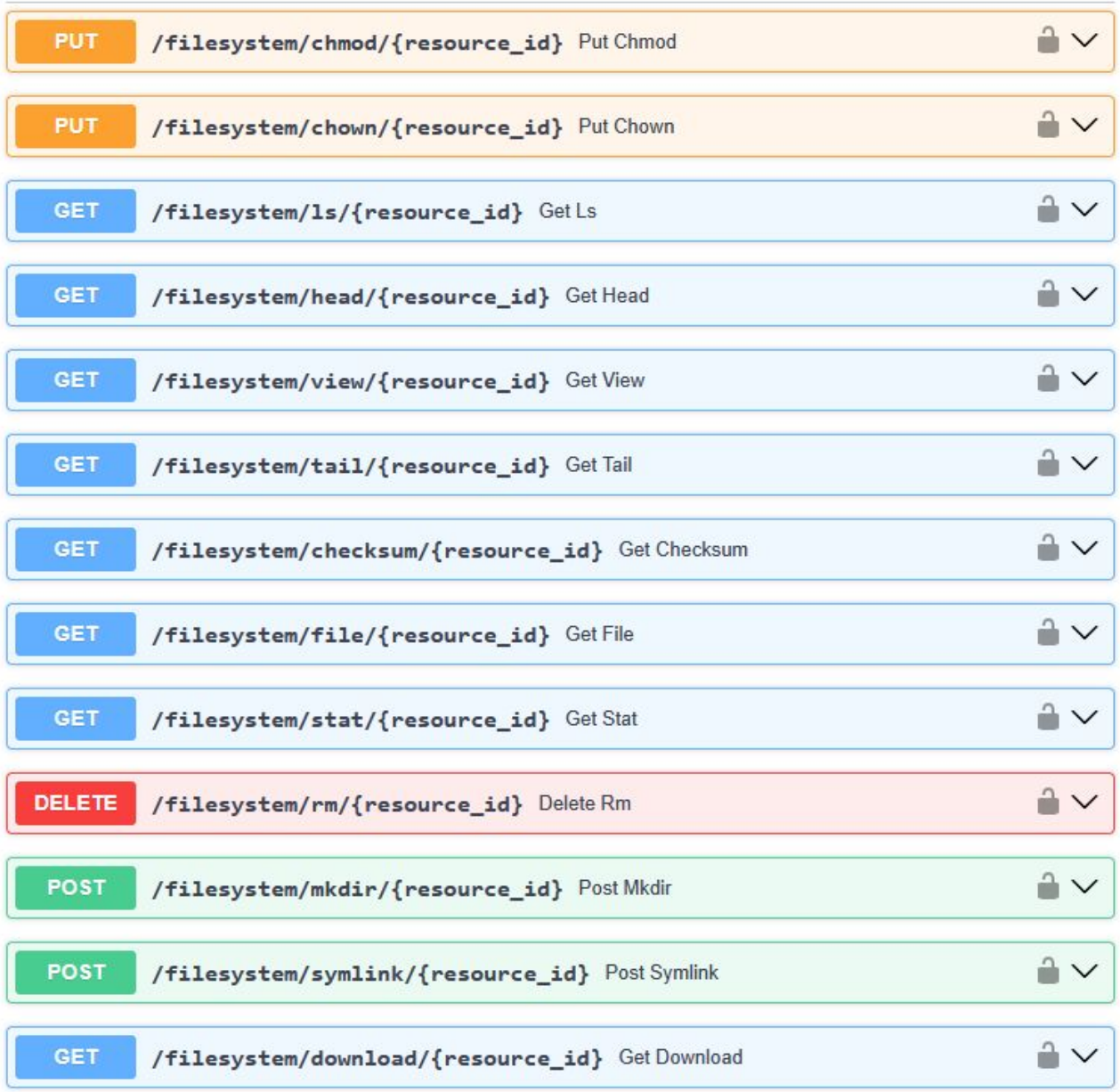


Roadmap – Job Submission

- PSI/J Initial Approach
- Supported operations
  - Submit
  - Live status
  - Delete
- V1 no Update/Alter
- Site-specific scripts
- Workload Managers
  - SLURM REST API
  - PBS GraphQL API

Roadmap - Local Filesystem Operations

- Local Operation Only
  - Complimentary to transfer frameworks
- Version 1: synchronous only
  - Async operations planned for later
- Similar to operations through SSH

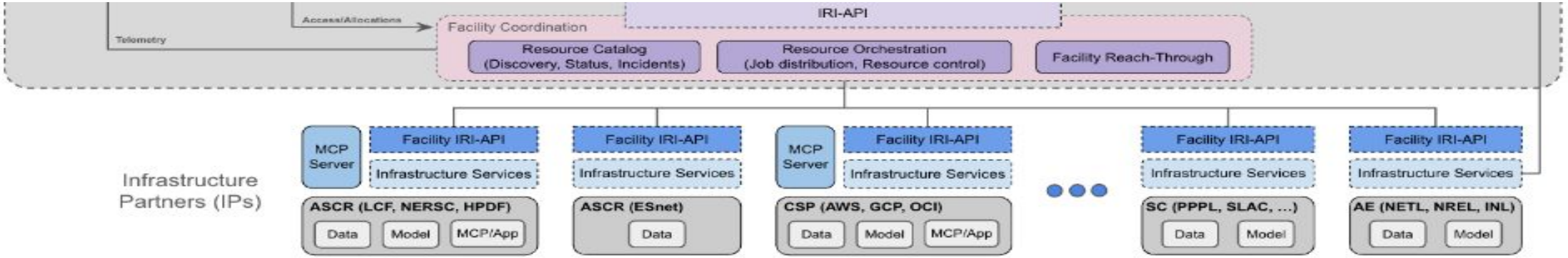


Roadmap

- Near Term
  - Containerized Workflows
    - K8s-based workflows targeted
  - Resource Orchestration
    - Cross-facility resource discovery and typing
    - Multi-facility Access Point
  - WAN Data Fabric Control
- Version 2 Features
  - Genericized Job Submission
  - Prearranged Workflows
  - Secure Enclave Support
  - Asynchronous Local Filesystem Operations

Roadmap – Resource Orchestration

- Higher level “roll” up of Facility and Partner IRI interfaces
- Orchestrator directs tagged work to appropriate resources
- Direct facility access still available
- Work being done in partnership with AmSC



Excerpt from “Introduction to the American Science Cloud”

Panel Q&A

Bjoern Enders
NERSC

John MacAuley
ESnet

Paul Rich
ANL

Juan Pablo Dorsch
CSCS

Tiffany Connors
NERSC

Eli Dart
ESnet

SC25 ST. LOUIS hpc ignites.

INVITED TALKS (FOR EXAMPLE)

Please Use Initial Caps in the Title of Your Talk or Session

Contributor Name, Contributor Affiliation; Second Contributor Name, Second Contributor Affiliation; Third Contributor Name, Third Contributor Affiliation; and so on...

This is an example of a formatted title slide.
Please delete this from your deck after viewing.

SC25 ST. LOUIS hpc ignites.